

BUNDESREPUBLIK DEUTSCHLAND



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 197 54 640.4

Anmeldetag: 09. Dezember 1997

Anmelder/Inhaber: Robert Bosch GmbH, 70469 Stuttgart/DE

Bezeichnung: Verfahren zur Koordination von
Netzwerkkomponenten

IPC: H 04 L, G 06 F

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 20. Dezember 2005
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Agurks

CERTIFIED COPY OF
PRIORITY DOCUMENT

Verfahren zur Koordination von Netzwerkkomponenten

STAND DER TECHNIK

- 5 Die vorliegende Erfindung betrifft ein Verfahren zur Koordination bzw. Abstimmung von Netzwerkkomponenten, insbesondere Master- und Slavekomponenten, wobei mindestens eine erste logische Komponente und eine zweite logische Komponente vorgesehen sind, welche jeweils einer bestimmten Applikation entsprechen und über das Netzwerk auf einer von der Applikationsebene im wesentlichen unabhängigen Kommunikationsebene miteinander kommunizieren können.

- 10 Die Erfindung ist für weite Bereiche der Kommunikationstechnik anwendbar, sie wird jedoch beispielshalber nur in bezug auf ein CAN-Netzwerksystem bei Anwendung im KFZ-Bereich näher erläutert.

- 20 Bei solchen, in einem KFZ installierten Systemen ergibt sich oftmals die Situation, daß nur einzelne Komponenten des Netzwerksystems nach einem Fehlerfall einen Neustart, d.h. Reset, durchführen. Ein typischer Auslöser für einen solchen Reset ist die Erkennung einer Unterspannungsbedingung, die von Komponente zu Komponente variieren kann.

25

Hieraus resultiert das grundsätzliche Problem, spezielle Kommunikationsmechanismen zwischen den Komponenten zu definieren, um möglichst schnell ein Wiederaufsetzen der verteilten Applikationen zu erreichen. Hierbei ist es im

allgemeinen vorteilhaft, den Zustand des Gesamtsystems vor dem Reset - soweit möglich - wieder zu reproduzieren bzw. abzustimmen. Von besonderer Bedeutung ist dies für Systeme mit Benutzerinteraktion, beispielsweise über ein
5 Anzeige- und Bedienteil, um durch Resets ausgelöste Störungen im Bedienablauf, wie etwa nicht mehr aktuelle Darstellungen auf einem Display, Zeitverzögerungen bei der Bedienung von Komponenten, verlorengegangene Eingaben etc. zu minimieren.

10

Die der vorliegenden Erfindung zugrunde liegende Problematik besteht also allgemein darin, in einem Netzwerk mit verteilten Applikationen die den Applikationen entsprechenden Netzwerkkomponenten auf effektive Weise zu koordinieren. Dies gilt insbesondere für einen Neustart nach
15 einem Fehlerfall von einzelnen Komponenten.

Generell müssen die folgenden vier Fälle erkannt und angemessen behandelt werden.

20

- i) "Normales" Hochfahren aller Komponenten (Systemstart),
- ii) zeitgleicher Fehler aller Komponenten nach Systemstart, z. B. durch Unterspannung für alle Komponenten im Netzwerk,
25
- iii) Fehler eines Teilsystems nach Systemstart, d. h. Fehlerbedingung einer oder mehrerer Komponenten, und

iv) Hardware-Reset nach erstmaligem Einschalten (First Power-On) oder fatalem Fehler.

5 Zunächst wird die spezielle Problematik einer gegenseitigen Koordination bzw. Synchronisation von verteilten Applikationen und deren diesen entsprechenden logischen Komponenten näher erläutert.

10 In Systemen, in denen Komponenten miteinander über ein Netzwerk, beispielsweise ein Bussystem, kommunizieren, ist eine gewisse Grundvoraussetzung für das hier beschriebene Verfahren eine Trennung von Kommunikation und Applikation innerhalb einer Komponente (s. ISO 7498, Information processing systems - Open systems Interconnection Basic Reference Model, 1984).

20 Betrachtet werden hier Systeme, die über ein Netzwerk, z.B. ein Bussystem, kommunizieren, wobei Kommunikation und Applikation folgendermaßen definiert sind.

25 Unter die Kommunikation fallen alle Funktionen, die zum gesicherten Datenaustausch mit anderen Komponenten benötigt werden. Typischerweise findet eine aus dem OSI-Modell der ISO (s.o.) abgeleitete Schichtung der Kommunikation Verwendung, d. h. eine Umsetzung von der physikalischen Schicht bis zur Applikationsschnittstelle. Den Anforderungen dieses Verwendungsgebietes angepaßt wird für den KFZ-Bereich in den gängigen Lösungen nur eine Un-

termenge des OSI-Modells eingesetzt, d.h. einige Schichten bleiben "leer". Weiterhin wird in Erweiterung des OSI-Modells üblicherweise ein die verschiedenen Schichten umfassendes Netzmanagement eingesetzt, welches die unterschiedlichen Komponenten hinsichtlich der Kommunikation synchronisiert.

Unter Applikation wird die spezifische Aufgabe jeder Komponente verstanden, z. B. die Funktionalität eines CD-Spielers oder die verschiedenen Funktionen eines Autotelefon.

In Systemen, die über logische Punkt-zu-Punkt(1:1)-Verbindungen zwischen den einzelnen Komponenten verfügen, werden über eine solche Verbindung die "Kommandos" der Applikation einer Komponente A an die Komponente B weitergeleitet, die diese dann z. B. mit einer Applikations-Rückmeldung beantwortet. Ein Beispiel ist die Aktivierung eines CD-Changers zum Spielen durch ein Bedienteil.

Zweckmäßigerweise werden diese 1:1 Verbindungen der Applikationsebene auf die 1:1 Verbindungen der Transportschicht (Schicht 4 im OSI-Modell) abgebildet. Während die 1:1-Transportverbindungen von beiden beteiligten Komponenten aufgebaut bzw. im Fehlerfall zurückgesetzt werden können, was einem symmetrischen Verhalten entspricht, gilt dies nicht für die Applikationsebene. Hier ist beispielsweise nur eine Komponente A - der "Master" - berechtigt, eine Komponente B - den "Slave" - zu steuern.

Insbesondere trifft dies für das Schalten der Hauptzustände des Slaves zu, wie z. B. "EIN" und "AUS".

5 Zweckmäßigerweise liegt einem solchen System ein Netzmanagement zugrunde, das ebenfalls zwischen Master- und Slave-Funktionalität unterscheidet. In diesem Fall ist normalerweise ein „Applikations-Master“ auch gleichzeitig ein „Netzmanagement-Master“. Daneben sind allerdings auch
10 durchaus Systeme denkbar, die beim Netzmanagement nur gleichberechtigte Stationen kennen, bei denen demnach die Master-/Slave-Unterscheidung auf die Applikationsebene beschränkt ist. Ein typisches Beispiel für letztgenanntes Netzmanagement-Verfahren ist unter dem Namen „dezentrales Netzmanagement“ auf CAN-Netzen im Karosseriebereich der
15 KFZ-Industrie zu finden.

Sofern nicht ausdrücklich anders aufgeführt, bezieht sich daher im folgenden die Bezeichnung „Master“ bzw. „Slave“ immer auf die Applikationsebene.

20

Ein relativ einfaches System besteht demnach aus einem Master und einem oder mehreren Slaves, mit jeweils einer 1:1 Verbindung zwischen Master und jedem Slave. Allerdings können auch komplexere Systeme, bestehend aus mehreren Masterkomponenten mit denselben oder auch unterschiedlichen Slaves, gebildet werden. Voraussetzung ist
25 hierbei nur, daß für jede logische Verbindung eindeutig festgelegt ist, welche Komponente Master und welche Slave ist. Auf diese Weise läßt sich dann ein hierarchisches

System aus Master, Submaster(s) und Slaves bilden, wie in der älteren Anmeldung DE 196 373 12 beschrieben.

5 Üblicherweise ist für eine Koordination der netzweiten Applikationen in erster Linie der Master verantwortlich. Für den Slave reicht es aus, den Reset eines Masters beispielsweise über einen Netzmanagementdienst zu erkennen.

10 Dies führt beispielsweise beim Slave zum Einleiten bestimmter Notlauffunktionen oder auch zum selbständigen Herunterfahren.

15 Der Master erkennt den Reset einer Slave-Komponente entweder durch zyklisches Abfragen des Slave-Status oder durch einen erneuten Kommunikationsaufbau, eingeleitet durch den Slave (z.B. Kommunikationssystem mit Netzmanagement und Transportprotokoll nach den älteren Anmeldungen DE 41 31 133 bzw. DE 196 373 12).

20 Als nachteilhaft bei dem obigen bekannten Ansatz hat sich die Tatsache herausgestellt, daß das zyklische Abfragen des Slave-Status durch den Master aufwendig und kommunikationsintensiv ist, da meist keine Änderung des Status vorliegt. Des weiteren ist dieser Abfragemechanismus inflexibel, da im allgemeinen nur bereits installierte Komponenten abgefragt werden.

Schließlich bringt ein erneuter Kommunikationsaufbau durch den Slave als Alternative zu dem obigen zyklischen

Abfragemechanismus den Nachteil mit sich, daß der Master außer der Information, daß die Kommunikation wieder zustandegekommen ist, keine Mitteilungen über die Ursache des Resets und/oder dessen Vorgeschichte, nämlich beispielsweise den vorherigen Applikationsstatus erhält.

Die Art des eigenen Resets kann vom Master erkannt werden, z.B. durch Eintrag von eigenen Statusinformationen in einen nichtflüchtigen Speicher, wie beispielsweise einem EEPROM, und durch Auswertung des Eintrags bei einem nachfolgenden Neustart.

Findet der Master z.B. den Eintrag "System gestartet und in Normalbetrieb" nach einem Reset vor, so kann er auf einen Neustart durch eine Fehlerbedingung schließen. Ist dagegen "System heruntergefahren" eingetragen, so handelt es sich um ein normales Hochfahren. Problematisch ist im Fehlerfall die begrenzte Speichermöglichkeit im Master.

Im allgemeinen kann nur der eigene Status gesichert werden, da für die Zustandsspeicherung aller angeschlossener Slaves die Zeit und/oder die Speicherkapazität normalerweise nicht ausreicht. Darüber hinaus ist ein Rückgriff auf beispielsweise in einem EEPROM abgespeicherte Statusinformationen über den Slave speziell beim Fehler-Reset des Masters risikoreich, da ja auch der Slave einen Reset durchgeführt haben könnte und damit der gespeicherte Slave-Zustand vom aktuellen Zustand abweicht.

Da in den obigen Standardlösungen keine detaillierten bzw. gesicherten Status-Informationen über den Slave vorliegen, wird der Master im allgemeinen die Slave-
5 Applikation neu starten, bzw. initialisieren.

Da hierdurch im allgemeinen vorherige Einstellungen zurückgesetzt werden müssen, bedeutet dies den Verlust des Wissens um die Vorgeschichte des Slaves, d.h. der ursprüngliche Betriebszustand oder Status der Applikation
10 ist nicht mehr ableitbar. Aus dem Neustart der Slave-Applikation resultieren im allgemeinen merkliche Verzögerungen.

15 VORTEILE DER ERFINDUNG

Das erfindungsgemäße Verfahren mit den Merkmalen des Anspruchs 1 weist gegenüber dem bekannten Lösungsansatz den Vorteil auf, daß es den alten Applikationszustand (Applikationsstatus), beispielsweise eines Slaves, vor Auftreten eines Fehlers an beliebiger Stelle im Netzwerk effizient wiederherstellen kann.
20

Dadurch wird es beispielsweise ermöglicht, gegebenenfalls nur lokal an dem betroffenen Slave einen Reset durchzuführen, der von dem Benutzer des Systems -beispielsweise einem Autofahrer - nicht wahrgenommen wird, da ein globales Reset ausbleiben kann.
25

Durch das gezielte Wieder-Aufsetzen eines kurzzeitig ausgefallenen Slaves ergibt sich im allgemeinen ein Zeitvorteil gegenüber üblichen Lösungen, bei denen alle laufenden Applikationen neu initialisiert werden müssen. Weiter ergeben sich zusätzliche Möglichkeiten für eine netzweite Diagnose.

10 Gemäß der Erfindung ist durch die Übermittlung der Vorgeschichte, vorzugsweise in Verbindung mit einer Netzwerkzustandstabelle in dem Master, auch ein sicheres Erkennen von neuen Komponenten im Netz ohne großen Konfigurationsaufwand für den Master möglich. Da in der erfindungsgemäßen Informationsnachricht, die im folgenden auch als

15 Start-up-Info bezeichnet wird, zweckmäßigerweise mitgeteilt wird, ob eine Komponente zum ersten Mal am Netzbetrieb teilnimmt, kann der Master entsprechend reagieren, z.B. durch das Freischalten von neuen Bedienmenüs.

20 Auch ermöglicht das erfindungsgemäße Verfahren die Verriegelung von Applikationen in bestimmten Zuständen, falls mehr als ein Master auf einen Slave zugreift. Dies gilt insbesondere für den Fall, daß ein Master den Slave in einen Diagnosemodus gebracht hat. In diesem Fall ist

25 der Slave im allgemeinen nicht mehr von einem weiteren Master steuerbar. Dies ist für diesen Master ebenfalls aus der Start-up-Info z.B. nach einem Reset unmittelbar ersichtlich. Die Start-up-Info ist daneben grundsätzlich für die Übermittlung von sogenannten Semaphoren, die

wichtig für die Synchronisation von verteilten Applikationen bzw. von Prozessen im DV-technischen Sinne sind, geeignet.

- 5 Die der vorliegenden Erfindung zugrundeliegende Idee besteht darin, das Wiederaufsetzen auf Applikationsebene durch den bzw. die Master unter Berücksichtigung des aktuell gemeldeten Applikationsstatus durchzuführen, indem der bzw. die Slaves ihren aktuellen Status in einem Speicher sichern, nach einer Änderung jeweils wieder aktualisieren und nach bestimmten Ereignissen, beispielsweise einem Fehler in der Sphäre des Slaves, eine Start-up-Info an den Master senden.
- 10
- 15 Hierbei wird als Auslöser letzterer Nachricht der erfolgreiche Aufbau der 1:1-Verbindung zwischen Master und Slave verwendet. Als vom Master ausgehende Trigger für diesen Verbindungsaufbau können wiederum zum einen bestimmte Netzmanagement-Mechanismen wie bestimmte Netzmanagement-
- 20 Telegramme, im folgenden auch als Watchdog-Telegramme bezeichnet, oder aber auch z.B. eine vom Master gesteuerte Schaltleitung Verwendung finden.

- Weiterhin kann dieser Trigger auch im Slave begründet
- 25 liegen. Ein typisches Beispiel bildet das Aufwecken eines Netzwerks durch eine Telefonkomponente. In diesem Fall geht die Initiative zum Verbindungsaufbau allein vom Slave aus.

In den Unteransprüchen finden sich vorteilhafte Weiterbildungen und Verbesserungen des in Anspruch 1 angegebenen Verfahrens.

5

Gemäß einer bevorzugten Weiterbildung werden die Informationen der übermittelten Informationsnachricht in der ersten logischen Komponente in dem nicht-flüchtigen Speicher gespeichert.

10

Gemäß einer weiteren bevorzugten Weiterbildung verwendet die erste logische Komponente die Informationen der übermittelten Informationsnachricht zur Rekonstruktion des Zustands vor dem Aufbauen der Kommunikationsverbindung.

15

Gemäß einer weiteren bevorzugten Weiterbildung enthält die Informationsnachricht Informationen bezüglich der Initiative zum Aufbauen der Kommunikationsverbindung.

20

Gemäß einer weiteren bevorzugten Weiterbildung stehen die erste und die zweite logischen Komponente in einer Master-Slave-Beziehung zueinander.

25

Gemäß einer weiteren bevorzugten Weiterbildung wird die Informationsnachricht nach einem Reset bei bereits gestarteter Applikation mindestens einer der ersten oder der zweiten logischen Komponente übermittelt.

Gemäß einer weiteren bevorzugten Weiterbildung enthält die Informationsnachricht Informationen darüber, ob die zweite Komponente einen Hardware-Reset durchgeführt hat.

- 5 Gemäß einer weiteren bevorzugten Weiterbildung werden bestimmte Informationen, die Inhalt der Informationsnachricht werden, in einem nichtflüchtigen Speicher der zweiten logischen Komponente gespeichert.

- 10 Gemäß einer weiteren bevorzugten Weiterbildung wird der Inhalt des nichtflüchtigen Speichers der zweiten logischen Komponente jedesmal dann aktualisiert, wenn eine Änderung des Applikationsstatus der zweiten logischen Komponente eintritt.

15

Gemäß einer weiteren bevorzugten Weiterbildung enthält die Informationsnachricht weitere Informationen zur Vorgeschichte der zweiten Komponente.

- 20 Besonders bevorzugte Anwendungen des erfindungsgemäßen Verfahrens liegen in Meß- und/oder Überwachungssystemen, in KFZ-Boardinformationssystemen und in netzwerkweiter Fehlerortung und/oder Fehlerdiagnose von Netzwerkkomponenten solcher Systeme.

25

ZEICHNUNGEN

Ausführungsbeispiele der Erfindung sind in den Zeichnungen dargestellt und in der nachfolgenden Beschreibung näher erläutert.

Es zeigen:

- 10 Fig. 1 eine schematische Darstellung der gemäß einer ersten Ausführungsform am erfindungsgemäßen Verfahren beteiligten Netzwerkkomponenten, nämlich einen Master, der mit einem Slave verbunden ist.
- 15 Fig. 2 den zeitlichen Verlauf des erfindungsgemäßen Verfahrens gemäß der ersten Ausführungsform zeitlich von oben nach unten fortschreitend bei einem Reset des Slaves in Folge einer Unter-
- 20 spannung.

BESCHREIBUNG DER AUSFÜHRUNGSBEISPIELE

- Fig. 1 zeigt eine schematische Darstellung der gemäß einer ersten Ausführungsform am erfindungsgemäßen Verfahren beteiligten Netzwerkkomponenten, nämlich einen Master,
- 25 der mit einem Slave verbunden ist.

In Fig. 1 bezeichnet Bezugszeichen 1 ein Radio, 2 einen CD-Changer, 3 eine Kommunikationsverbindung zwischen dem

Radio 1 und dem CD-Changer 2 sowie 4 und 5 jeweils nicht-flüchtige Speicher.

Das relativ einfache System enthält das Radio 1 als Master und den angeschlossenen CD-Changer 2 als Slave des Radios 1 und eine die beiden Netzwerkkomponenten 1, 2 miteinander verbindende Kommunikationsverbindung 3 in Form einer Busleitung. Weiter verfügen sowohl das Radio 1 als auch der CD-Changer 2 über nichtflüchtige, Speicher 4 bzw. 5, die beispielsweise in Form von gepufferten SRAMs oder EEPROMs realisiert sein könnten. Die Speicher 4, 5 dienen der Speicherung von Daten, deren Auswertung durch den Master 1 zur Koordination der Netzwerkkomponenten 1, 2 dienen.

15

Figur 2 zeigt schematisch den zeitlichen Verlauf des erfindungsgemäßen Verfahrens gemäß der ersten Ausführungsform zeitlich von oben nach unten fortschreitend bei einem Reset des Slaves in Folge einer Unterspannung.

20

In Fig. 2 bezeichnet 6 ein Watchdog-Telegramm, d.h. eine Aufforderung des Radios 1 an den CD-Changer 2, eine Kommunikation aufzubauen bzw. aufrechtzuerhalten.

25 Die linke äußere Linie stellt die Applikationsschnittstelle 1A des Radios 1 dar, die zweite Linie von links die Kommunikationsschnittstelle 1K des Radios 1, die dritte Linie von links die Kommunikationsschnittstelle 2K

des CD-Changers 2 und die rechte äußere Linie die Applikationsschnittstelle 2A des CD-Changers 2.

5 Anfangs und in der Figur oberhalb von t_1 ist das System initialisiert und beide Komponenten, das Radio 1 und der CD-Changer 2, sind in Betrieb. Im CD-Changer 2 ist dessen Status, beispielsweise „PLAY“, in dem nichtflüchtigen Speicher 5 gespeichert. Vom Radio 1 werden zyklisch Watchdog-Telegramme 6 an den CD-Changer 2 gesendet.

10

Zum Zeitpunkt t_1 selbst tritt eine Betriebsstörung, hier eine Unterspannung im CD-Changer 2 auf, die einen Reset des CD-Changers 2 nötig macht. Sobald diese Unterspannung als solche erkannt ist, werden neben dem bereits gespeicherten gegenwärtigen Applikationszustand „PLAY“ des CD-Changers 2 erwünschte weitere Informationen über seine Vorgeschichte, z.B. „Fehler im Slave“ sowie die letzte Position des CD-Abtastsystems, in seinem nicht-flüchtigen Speicher 5 gespeichert.

20

Dazu sei ergänzend erläutert, daß bei Auftreten einer Unterspannung bei einer Komponente die Versorgungsspannung i.a. nicht auf Null Volt heruntergeht. Grund dafür ist eine Pufferung der Versorgungsspannung. Bei leichten Unterspannungen können bestimmte Maßnahmen zur Sicherung aktueller Daten in den nicht-flüchtigen Speicher (soweit vorhanden) eingeleitet werden. Damit verbunden sind normalerweise ein Zurücksetzen der Applikation und/oder der Kommunikation (z.B. Abbau logischer Verbindungen). Steigt

. . .

die Spannung wieder an, so müssen Applikation sowie ggfs. Kommunikation neu initialisiert werden („Warmstart“).

5 Dabei bedeutet Initialisierung der Applikation eine Vorbereitung der Applikation und ggfs. einen Start der Applikation (i.a. kein selbständiger Applikationsstart bei einem Slave). Initialisierung der Kommunikation bedeutet das Treffen von Vorbereitungen für die Kommunikation mit anderen Netzwerkkomponenten, z.B. einen Verbindungsauf-
10 bau.

Erst wenn die Versorgungsspannung unter einen kritischen Wert sinkt droht ein Hardware-Reset des Mikrocontrollers sowie der Verlust aller Daten im flüchtigen Speicher
15 (RAM). Nach einem Hardware-Reset muß die Komponente einen sogenannten „Kaltstart“ durchführen, bei dem i.a. eine spezielle Initialisierungsroutine durchlaufen werden muß. Solch eine Grundinitialisierung betrifft die Komponente auf der untersten Ebene, z.B. Setzen/Zurücksetzen von µC-
20 Ports usw. Wird in dieser Routine beispielsweise ein Flag im RAM gesetzt, besteht die Möglichkeit, einen Kaltstart (Verlust aller Daten, wenn kein nicht-flüchtiger Speicher zur Verfügung steht) von einem Warmstart (Teilverlust von Daten) zu unterscheiden. Der Begriff Reset ist im folgen-
25 den also nicht zwangsläufig mit einem Hardware-Reset des Mikrocontrollers gleichzusetzen.

Der CD-Changer 2 erhält nun bei t_2 ein Watchdog-Telegramm 6 vom Radio 1 und baut daraufhin die Verbindung zum Radio 1 wieder auf.

- 5 Bei t_3 quittiert das Radio 1 dem CD-Changer 2 den Verbindungsaufbau. Hiermit verbunden ist eine Reinitialisierung der Kommunikationsverbindung im Radio 1.

- 10 Bei t_4 sendet der CD-Changer 2 daraufhin die Informationsnachricht, die Start-Up-Info, an das Radio 1. Das Radio 1 verarbeitet diese Informationsnachricht und sendet infolgedessen bei t_5 geeignete Applikationsnachrichten zur Wiederherstellung des alten Zustands vor dem Auftreten der Unterspannung an den CD-Changer 2, beispielsweise
15 die Befehle "Einschalten" und "Play". Somit ist der Zustand wiederhergestellt, in dem das System vor Eintritt der betriebsstörenden Unterspannung war.

- 20 Zur Begründung für den hier beschriebenen Ablauf sei bemerkt, daß es im Regelfall nicht sinnvoll ist, daß der Slave versucht, selbständig seinen alten Zustand komplett zu restaurieren. Hintergrund ist, daß auch im Master (oder in einer anderen, hier nicht gezeigten Komponente) ein Fehler aufgetreten sein kann, so daß die verschiedenen
25 Applikationen nicht mehr synchronisiert sind. Beispielsweise würde es keinen Sinn ergeben, wenn der CD-Changer selbständig seinen alten Zustand „PLAY“ restaurieren würde, das Radio aber aufgrund eines internen Re-

sets die Audio-Signale des CD-Changers nicht mehr verarbeiten würde.

Der wesentliche Inhalt der Informationsnachricht "Start-
5 up Info" ist der aktuelle Applikationszustand des CD-
Changers 2 bzw. des Slaves, soweit er für die Bedienung
des CD-Changers 2 aus Sicht des Radios 1 bzw. des Masters
von Bedeutung ist. Bedeutsame Applikationshauptzustände
aus Sicht dieses Masters sind beispielsweise EIN/AUS,
10 d.h. sie zeigen dem Radio 1 an, ob die Slave-Applikation
für den Master verfügbar ist oder nicht.

Es können jedoch auch andere, spezielle Applikationszu-
stände aktiviert gewesen und dementsprechend nun dem Ma-
15 ster angezeigt werden, wie beispielsweise die Wiedergabe
eines CD-Changers (PLAY) im vorliegenden Ausführungsbei-
spiel oder etwa ein Diagnose-Modus.

Weitere spezielle Zustände des Slaves können kennzeich-
20 nen, daß der Slave nur einen eingeschränkten Funktionsum-
fang besitzt, da beispielsweise die Applikationssoftware
noch von einer CD-ROM geladen werden muß oder weil eine
Verriegelung des Slaves seitens anderer logischer Verbin-
dungen besteht, wie oben bereits erwähnt wurde.

25 Desgleichen können diverse Zustände der Mensch-Maschine-
Schnittstelle des Slave, also beispielsweise Menüs oder
Fenster auf einem Bildschirm, die aktuelle Belegung von

Softkeys im System, dem Master auf eine solche Weise angezeigt werden.

5 Ein optionaler Bestandteil ist der sogenannte Initialisierungsanstoß. Dies ist ein Bestandteil, der angibt, wer letztlich den Aufbau der Kommunikationsverbindung initiiert hat.

10 Dieser Anstoß kann sowohl im Master als auch im Slave begründet liegen. Im ersten Fall, der den Normalfall darstellt und dem obigen Beispiel zugrunde liegt, liefert der Master den Initialisierungsanstoß, z.B. durch einen Netzmanagement-Mechanismus in Form eines Watchdog-Telegamms. Ein Beispiel für den zweiten Fall bildet das
15 Wecken durch einen Slave, welches beispielsweise durch die Verbindungsaufbautelegramme des Slaves ausgelöst wird. Aufgrund der übermittelten Information „Initialisierungsanstoß durch Slave“ kann der Master im letzteren Fall den unerwarteten Verbindungsaufbau durch den Slave
20 leicht als Wecken interpretieren und von Fehlerfällen unterscheiden.

Ein weiterer optionaler Bestandteil beschreibt die Vorgeschichte der Applikation. In diesem Fall wurde die Sitzung durch einen Fehler beendet, es kam zu einem Reset
25 der Kommunikation sowie der Applikation wegen des Auftretens einer Unterspannung des CD-Changers 2. Daher hat der Bestandteil die Bedeutung "Fehler im Slave" und kann

durch ein Flag oder ein anderes Kennzeichen signalisiert werden.

5 Wenn in einem anderen Fall die Applikation bei der letzten Sitzung, d.h. dem letzten zusammenhängendem Zeitraum mit einer gültigen 1:1-Verbindung zwischen Radio 1 und CD-Changer 2 beispielsweise durch einen Ausschaltbefehl ordnungsgemäß verlassen wurde, könnte der Inhalt des Bestandteils die Bedeutung "o.k." haben.

10

Befindet sich der CD-Changer 2 zum ersten Mal am Netz, dann könnte der Inhalt des Bestandteils die Bedeutung "NEU" haben. Ein solches Kennzeichen ist wichtig für eine Grundinitialisierung nach einem Austausch, einem Neueinbau oder zur Diagnose nach einem fatalen Fehler.

15

Bei der Übermittlung der Vorgeschichte ist zu beachten, daß die Start-up-Info ausschließlich die Sicht des Slaves widerspiegelt. Erst ein Vergleich mit den im Master gespeicherten Daten erlaubt eine endgültige Aussage zur Vorgeschichte des Systems.

20

Generell gilt nun, daß nur in der Komponente, die einen Reset während des Betriebs durchführt, Applikation und Kommunikation gleichermaßen initialisiert werden. In der verbundenen Komponente, hier dem Radio 1 als Master, wird dagegen nur ein Reset der Kommunikation ausgeführt. Der Reset ist nötig, um die 1:1-Kommunikationsverbindung wieder zu synchronisieren. Die Applikation des Masters ist

25

normalerweise nicht unmittelbar vom Reset der anderen Komponente betroffen.

5 Eine Initialisierung der Kommunikation ist also auch ohne eine Initialisierung der Applikation möglich. Bei einer 1:1-Verbindung erfolgt im Fehlerfall ein Reset der Kommunikationsschichten also auf beiden Seiten, während die Applikation beispielsweise nur auf einer Seite zurückgesetzt wird.

10

Das Verfahren beruht also auf einer automatischen Meldung des Slaves an den Master, die den aktuellen Zustand der Slave-Komponente sowie deren Vorgeschichte enthält, initiiert durch ein spezielles, eindeutiges Kommunikationsereignis. Dieses kann ausgelöst sein durch das Auftreten der Unterspannung am CD-Changer 2 bzw. durch den Reset des CD-Changers 2.

15

Aufgrund aller Start-up-Informationen der angeschlossenen Slaves sowie der eigenen, im nichtflüchtigen Speicher 4 des Masters enthaltenen Daten kann der Master 1 nun sehr effizient den alten Applikationszustand nach einem Fehler rekonstruieren.

20

25 Meldet eine Komponente abweichend vom Ausführungsbeispiel, daß sie neu im Netz ist, so zeigt ein Vergleich mit den gespeicherten Daten des Masters, ob diese Komponente einen Fehler, beispielsweise eine extreme Unterspannung, einen fatalen Softwarefehler oder dgl. hatte,

was beispielsweise wichtig für eine Systemdiagnose wäre, oder ob sie wirklich neu im Netz ist. In diesem Fall könnte ein spezieller Konfigurationslauf erfolgen, der in einem Ausführungsbeispiel weiter unten erläutert wird.

5

Zusammenfassend ermöglicht das erfindungsgemäße Verfahren gemäß diesem Ausführungsbeispiel also eine ereignisorientierte Merk- oder Speicherfunktion der Komponenten und enthält folgende Mechanismen.

10

Ein Slave sendet automatisch jedem ihm zugeordneten Master unmittelbar nach dem erfolgreichen (Wieder-)Aufbau der 1:1 Verbindung zum Master als erste Applikationsnachricht eine spezielle Informationsnachricht mit Statusinformationen ("Start-Up-Info") in den folgenden Fällen:

15

- nach erfolgter eigener Initialisierung bzw. eigenem Reset der Kommunikationsschichten und
- 20 - nach erfolgtem Reset der Kommunikationsschichten durch den Master.

20

25

Dabei ist zu bedenken, daß grundsätzlich beide - Master oder Slave - unabhängig von ihrer hierarchischen Einbettung in das Gesamtsystem des Netzwerks die gegenseitige Kommunikation beginnen können.

Da die Vorgeschichte anhand der Start-Up Infos der Slaves vom Master abgeleitet werden kann, ergeben sich zusätzliche Möglichkeiten für eine netzweite Diagnose.

- 5 Im folgenden zweiten Ausführungsbeispiel sei ein weiteres System angenommen, das aus einer beliebigen Master- und zwei und in einer weiter unten beschriebenen Abwandlung drei beliebigen Slave-Komponenten besteht.

10 Zunächst befinden sich neben dem Master zwei weitere Komponenten A, C - Slaves des Masters - im Netzwerk. Eine weitere Komponente B ist dem Master zwar bekannt, sie ist jedoch (noch) nicht angeschlossen.

- 15 Das erfindungsgemäße Verfahren läßt sich für dieses System auch in Anlehnung an Fig. 2 beschreiben.

Es wird eine in dem nichtflüchtigen Speicher des Masters abgelegte Netzwerkstatustabelle mit Konfigurations-, Verbindungs- und Applikationszuständen der bekannten Slaves
20 exemplarisch gezeigt.

Bekannte Slaves	Netz- konfiguration	Verbindungs- status	Applikations- status
Komponente A	"vorhanden"	"Verbindung aufgebaut"	"Applikation ein"
Komponente B	"nicht verfü- bar"	"keine Verbin- dung"	"nicht verfü- bar"
Komponente C	"vorhanden"	"Verbindung aufgebaut"	"Applikation aus"

Die Netzwerkstatustabelle wurde zum Zwecke einer besseren
5 Verständlichkeit so einfach wie möglich gehalten. Norma-
lerweise enthält eine solche Tabelle weitaus mehr Infor-
mationen, z. B. Konfigurationsdaten für die einzelnen
Verbindungen.

10 Aus Übersichtsgründen nennt dieses Beispiel nur die Ap-
plikationshauptzustände "EIN" und "AUS" bzw. "nicht ver-
fügbar". Neben den Informationen in dieser Netzwerkzu-
standstabelle speichert der Master auch seine eigene Vor-
geschichte.

15

In einer Abwandlung des letztgenannten Beispiels (Neu-
anschluß eines Slaves) erhält der Master nach einem Ver-
bindungsaufbau folgende "Start-up Info" von der Komponen-
te B:

20

B an Master: "B zum ersten Mal im Netz, Applikationszustand ,AUS' ".

- 5 Der Master kann aufgrund dieser Information einen speziellen Konfigurationslauf starten. Die Netzwerkstatustabelle ändert sich dann wie folgt:

Bekannte Slaves	Netz- konfiguration	Verbindungs- status	Applikations- status
Komponente A	"vorhanden"	"Verbindung aufgebaut"	"Applikation ein"
Komponente B	" vorhanden"	"Verbindung aufgebaut"	"Applikation aus"
Komponente C	"vorhanden"	"Verbindung aufgebaut"	"Applikation aus"

10

Von diesem Zeitpunkt an ist der Master in der Lage, die Applikation der Komponente B zu starten und alle verfügbaren Funktionen zu steuern.

- 15 Im folgenden wird ein beispielhafter Fehlerfall dargestellt.

- 20 In einer weiteren Veränderung des letztgenannten Systemzustands erhält der Master nach erneutem Verbindungsaufbau der Komponenten A und C folgende "Start-up Informationen":

A an Master: "A war bereits im Netz, Applikation ,AUS'".

5 C an Master: "C ist zum ersten Mal im Netz, Applikation
,AUS'".

10 Nach Vergleich mit den gespeicherten Informationen kann
der Master nun unmittelbar den alten Zustand durch ge-
zielte Applikationsnachrichten wieder rekonstruieren (in
diesem einfachen Beispiel ist nur eine Nachricht notwen-
dig):

Master an A: "Applikation ,EIN'".

15 Weiterhin kann der Master feststellen, daß Komponente A
einen "Warmstart" und Komponente C sogar einen "Kalt-
start" durchgeführt hatte, da Komponente C im Netz be-
reits vorhanden war. Diese Daten können mit speziellen
Fehlerzählern verwaltet und über Diagnosesoftware ausge-
20 wertet werden.

Obwohl die vorliegende Erfindung vorstehend anhand bevor-
zugter Ausführungsbeispiele beschrieben wurde, ist sie
darauf nicht beschränkt, sondern auf vielfältige Weise
25 modifizierbar.

So kann das erfindungsgemäße Verfahren auch für komplexe-
re Systeme eingesetzt werden, in denen der Master eines

oder mehrerer Slaves wiederum der Slave eines Masters auf einer ranghöheren Ebene ist.

Ebenso ist es einsetzbar in Systemen, in denen ein oder
5 mehrere Slaves mehr als nur einem Master zugeordnet sind.

Ebenso kann das erfindungsgemäße Verfahren in Netzwerken mit Client-Server-Architekturen verwendet werden, wobei i.a. der Client eine gewisse Masterfunktionalität übernimmt.
10

Desweiteren ist es prinzipiell in Überwachungs- und Fehlerdiagnosesystemen in beliebigen technischen, vernetzten Einrichtungen einsetzbar. Insbesondere im KFZ-Bereich ergeben sich dabei vielfältige Anwendungsmöglichkeiten bei
15 der Fehlerüberwachung und Fehlerdiagnose von vernetzten elektronischen Steuergeräten.

Ebenso können die Kommunikationsverbindungen zwischen den
20 Komponenten durch drahtlose Verbindungen realisiert sein.

Verfahren zur Koordination von Netzwerkkomponenten

PATENTANSPRÜCHE

5

1. Verfahren zur Koordination von Netzwerkkomponenten (1, 2), wobei mindestens eine erste logische Komponente (1) und eine zweite logische Komponente (2) vorgesehen sind, welche jeweils einer bestimmten Applikation entsprechen und über das Netzwerk auf einer von der Applikationsebene im wesentlichen unabhängigen Kommunikationsebene miteinander kommunizieren können, mit den Schritten:

10

15

Aufbauen einer Kommunikationsverbindung zwischen der ersten und der zweiten logischen Komponente (1, 2) auf Initiative einer der ersten oder zweiten logischen Komponente (1, 2) ansprechend auf ein bestimmtes Ereignis, welches die erste und/oder zweite logische Komponente (1, 2) betrifft;

20

25

Übermitteln einer Informationsnachricht über die aufgebaute Kommunikationsverbindung von der zweiten logischen Komponente (2) an die erste logische Komponente (1), wobei die Informationsnachricht zumindest Informationen bezüglich des aktuellen Applikationsstatus der zweiten logischen Komponente enthält; und

Vergleichen der Informationen der übermittelten Informationsnachricht in der ersten logischen Komponente (1) mit dort in einem nicht-flüchtigen Speicher (4) gespeicherten entsprechenden Informationen.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Informationen der übermittelten Informationsnachricht in der ersten logischen Komponente (1) in dem nicht-flüchtigen Speicher (4) gespeichert werden.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die erste logische Komponente (1) die Informationen der übermittelten Informationsnachricht zur Rekonstruktion des Zustands vor dem Aufbauen der Kommunikationsverbindung verwendet.
4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Informationsnachricht Informationen bezüglich der Initiative zum Aufbauen der Kommunikationsverbindung enthält.
5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die erste und die zweite logischen Komponente (1, 2) in einer Master-Slave-Beziehung zueinander stehen.

6. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Informationsnachricht nach einem Reset bei bereits gestarteter Applikation mindestens einer der ersten oder der zweiten logischen Komponente (1, 2) übermittelt wird.
7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß die Informationsnachricht Informationen darüber enthält, ob die zweite Komponente (2) einen Hardware-Reset durchgeführt hat.
8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß bestimmte Informationen, die Inhalt der Informationsnachricht werden, in einem nichtflüchtigen Speicher (5) der zweiten logischen Komponente (2) gespeichert werden.
9. Verfahren nach Anspruch 8, gekennzeichnet durch den Schritt, daß der Inhalt des nichtflüchtigen Speichers (5) der zweiten logischen Komponente (2) jedesmal dann aktualisiert wird, wenn eine Änderung des Applikationsstatus der zweiten logischen Komponente (2) eintritt.
10. Verfahren nach Anspruch 8 oder 9, dadurch gekennzeichnet, daß die Informationsnachricht weitere Informationen zur Vorgeschichte der zweiten Komponente (2) enthält.

11. Durchführung des Verfahrens nach einem der vorstehenden Ansprüche in einem hierarchisch aufgebauten System von Netzwerkkomponenten, von denen die erste logische Komponente (1) für eine oder mehrere zweite logische Komponenten (2) eine Masterfunktion ausübt.
5
12. Durchführung des Verfahrens nach Anspruch 11 in einem hierarchisch aufgebauten System von Netzwerkkomponenten, von denen die zweite logische Komponente (2) für eine oder mehrere erste oder zweite logische Komponenten (1, 2) eine Masterfunktion ausübt.
10
13. Anwendung des Verfahrens nach einem der vorstehenden Ansprüche in Meß- und/oder Überwachungssystemen und/oder Steuerungssystemen in der Fertigungstechnik.
15
14. Anwendung des Verfahrens nach einem der vorstehenden Ansprüche in KFZ-Bordinformationssystemen.
20
15. Anwendung des Verfahrens nach einem der vorstehenden Ansprüche zur netzwerkweiten Fehlerortung und/oder Fehlerdiagnose von Netzwerkkomponenten (1,2).
25

Verfahren zur Koordination von Netzwerkkomponenten

ZUSAMMENFASSUNG

5

Es wird ein Verfahren zur Koordination von Netzwerkkomponenten (1, 2) vorgeschlagen, wobei mindestens eine erste logische Komponente (1) und eine zweite logische Komponente (2) vorgesehen sind, welche jeweils einer bestimmten Applikation entsprechen und über das Netzwerk auf einer von der Applikationsebene im wesentlichen unabhängigen Kommunikationsebene miteinander kommunizieren können, mit den Schritten: Aufbauen einer Kommunikationsverbindung zwischen der ersten und der zweiten logischen Komponente (1, 2) auf Initiative einer der ersten oder zweiten logischen Komponente (1, 2) ansprechend auf ein bestimmtes Ereignis, welches die erste und/oder zweite logische Komponente (1, 2) betrifft; Übermitteln einer Informationsnachricht über die aufgebaute Kommunikationsverbindung von der zweiten logischen Komponente (2) an die erste logische Komponente (1), wobei die Informationsnachricht zumindest Informationen bezüglich des aktuellen Applikationsstatus der zweiten logischen Komponente enthält; und Vergleichen der Informationen der übermittelten Informationsnachricht in der ersten logischen Komponente (1) mit dort in einem nicht-flüchtigen Speicher (4) gespeicherten entsprechenden Informationen.

(Fig. 1)

. . .

Verfahren zur Koordination von Netzwerkkomponenten

BEZUGSZEICHENLISTE

5

1	erste, logische Komponente, Master (Radio)
2	zweite logische Komponente, Slave (CD- Changer)
3	Kommunikationsverbindung
4	nichtflüchtiger Speicher
5	vorzugsweise nichtflüchtiger Speicher
6	Watchdog-Telegramm

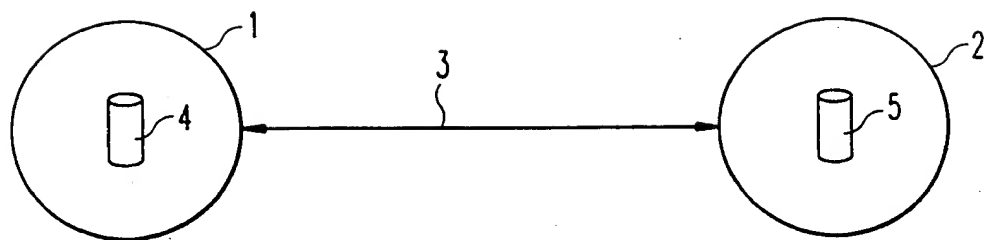


Fig. 1

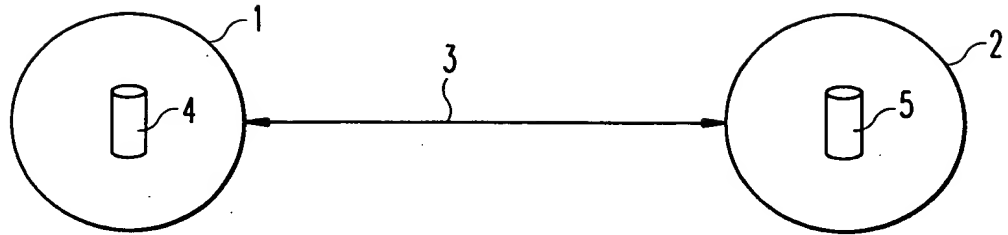


Fig. 2

